Performance Evaluation of AREA-MAC : A Cross-Layer Perspective

Pardeep Kumar, Mesut Güneş, Qasim Mushtaq, Jochen Schiller

Institute of Computer Science
Freie Universität Berlin, Berlin, Germany
{pardeep.kumar, mesut.guenes, qasim.mushtaq, jochen.schiller}@fu-berlin.de

ABSTRACT

A medium access control (MAC) protocol generally regulates the access of devices to a shared medium. In case of wireless sensor networks (WSNs), it is also responsible to create energy-efficient links between nodes, where messages can be sent to the sink node in a timely manner. This paper evaluates AREA-MAC, a medium access control protocol designed for real-time and energy-efficient WSN applications. AREA-MAC uses the low power listening (LPL) technique with short preamble messages to minimize latency, energy consumption, and control overhead on nodes. Though AREA-MAC is mainly associated with the MAC layer, it exploits simple routing at the network layer and interacts with application as well as PHY layers. In this evaluation a cross-layer design perspective is adopted, where direct interaction between different layers is examined. Through simulation study, we evaluate the timeliness, energy-efficiency, and packet reception ratio for nodes using AREA-MAC with and without cross-layer support. Results show that the cross-layering improves network performance in terms of timeliness and energy-efficiency but it comes at the cost of relatively low packet reception ratio at the sink node.

Keywords: WSN, MAC, real-time, energy, packet reception ratio

1 INTRODUCTION

Various MAC protocols [1–9] proposed for WSNs are mainly designed to accomplish the major objective of energy saving. Factors like latency, adaptivity to traffic conditions, packet reception, and system fairness are mostly ignored or dealt with as secondary objectives. With the increased interests for WSNs in medical, security, monitoring, and home automation applications, provision of real-time guarantees is as crucial as prolonging their lifetime.

Furthermore, most of the proposed protocol follow the traditional layered architecture, where they try to improve performance only at the respective layer. With very limited resources available for WSNs, cross-layer design could boost their overall performance [10–13]. Unlike layered networks, WSNs can not afford significant layered overhead due to their limited energy, storage, and processing capabilities. Moreover, application-aware communication and low-power radio considerations motivate for the cross-layer architecture for WSNs.

An *A*synchronous *R*eal-time *E*nergy-efficient and *A*daptive MAC (AREA-MAC) protocol [14] provides an improved performance in terms of timeliness and energy-efficiency and maintains an acceptable trade-off between other parameters. AREA-MAC uses the LPL technique [9] with short preamble messages. Other MAC protocols like WiseMAC [4] and B-MAC [9] use the LPL with long preambles, where nodes remain awake for the whole preamble time even if they are not the target node. This cause higher latency, energy consumption, and control overhead on nodes [8]. However, nodes using AREA-MAC have short and adaptive preambles with a destination address and an acknowledgement combination. A node using AREA-MAC wakes up very shortly to sense the carrier activity and sends a short preamble packet prior to the data packet, if the channel is free. The node then waits for a short period of time to receive an acknowledgement from the next hop. A data packet is immediately sent to the next hop on reception of an acknowledgement from the next-hop node. On the other hand, if the node does not receive an acknowledgement within the specified time period, it goes to sleep mode for a very short time and wakes up again. This process minimizes latency and energy consumption not only on source node but also on non-targeted nodes in the shape of reduced collisions, idle listening, overhearing, and over-emitting. The detailed working of AREA-MAC is discussed in the next section.

AREA-MAC exploits network, physical, and application layers for simple routing, radio and channel status, and application aware traffic generation respectively. The comparison of AREA-MAC with the B-MAC protocol is shown in [14], where AREA-MAC outperforms B-MAC in almost every aspect. In this paper, we evaluate the performance of AREA-MAC with the cross-layer prospective. We examine the effect of three different types of routings at the network layer and two types of traffic generating scenarios at the application layer. Several results shown with and without routing and burst traffic generation clarify that the cross-layering improves AREA-MAC performance in terms of timeliness and energy-efficiency but it marginally reduces the packet reception ratio at the sink node.

The remainder of this paper is organized as follows. The next section details the working of AREA-MAC with its characteristics, network, energy, and delay models. Subsequently, in Section 3, cross-layering with AREA-MAC is discussed. Afterwards, in Section 4, the evaluation of AREA-MAC is elaborated. Section 5 briefs state of the art, whereas Section 6 concludes the paper.

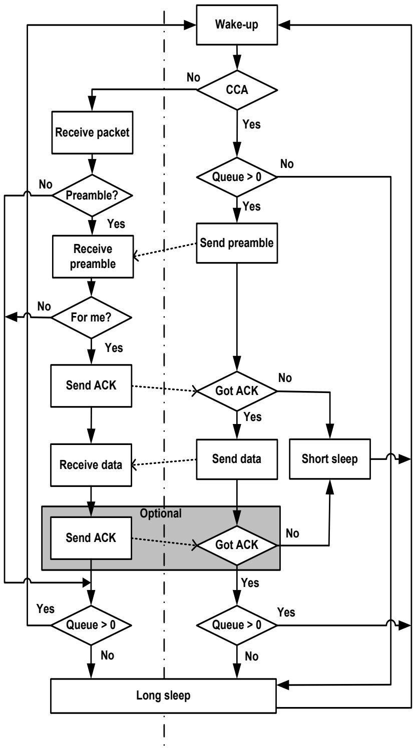

Figure 1: A flow chart showing reception and transmission (left and right side of the dotted vertical line respectively) for a node using AREA-MAC.

2 AREA-MAC

AREA-MAC is designed to provide a suitable solution for time critical and energy-efficient WSN applications and, at the same time to provide an acceptable trade-off between other parameters. Nodes using AREA-MAC have short and adaptive preambles with a destination address and an acknowledgement combination. A node using AREA-MAC wakes-up very shortly to sense the carrier activity. It sends a short preamble packet prior to the data packet, if the channel is free. A preamble packet contains source and next hop addresses. The node then waits for a short period of time to receive an acknowledgement from the next hop. A data packet is sent to the next hop only on reception of an acknowledgement from the next-hop node. Otherwise, the source node goes to the **short sleep** mode, wakes-up after a very short time interval, and tries again. The node goes to the **long sleep** mode after failing to receive an acknowledgement for a maximum number of allowed attempts. The short sleep mode is used to facilitate real-time applications, where nodes go to sleep mode for a very short period of time. This time is little longer than the time required to switch the radio from one mode to other. However, the long sleep is the normal time for which nodes have to sleep periodically. This depends on the slot duration time selected by the application.

On the other hand, if the channel is busy, the node tries to receive a preamble. If the next hop address of the preamble matches with its address, it immediately sends an acknowledgement and receives data from the source node. Otherwise it goes back to sleep mode. This whole process of reception and transmission of a node using AREA-MAC is sketched in Figure 1. In order to increase the data reliability, the optional data-acknowledgement mechanism (colored box in Figure 1) may be used where, after sending a data packet, a node waits for an acknowledgement from the source node. If it does not receive an acknowledgement within an specific time, it goes to a short sleep mode, wakes up soon, and repeats the whole process again. Of course, this reliability will effect network timeliness and energy consumption.

In order to further improve timeliness, nodes after completing their reception or transmission, check their data queue rather than going to sleep mode directly. If there is a data packet in its data queue, the node starts its cycle directly from the wake-up mode (see Figure 1).

2.1 Characteristics

The main characteristics of AREA-MAC include asynchrony, energy-efficiency, timeliness, and adaptivity to traffic conditions. Nodes using AREA-MAC are fully independent of sleep and wake-up schedules of other nodes. They do not require a system-wide synchronization, which resolve overhead and scaling problems. AREA-MAC is also an energy-efficient protocol, where nodes perform duty-cycle and wake up very shortly to check the channel activity without actually receiving any data. They go back to sleep mode if the channel is idle; otherwise they try to receive a preamble, if available. On reception of a preamble, the node matches its address with the next hop address of the preamble. On a successful match, a node acknowledges the source node immediately, which causes the source node to stop sending further preambles and to start transmitting data packets. All non-target nodes go back to sleep mode immediately which minimizes the possibility of a collision, idle listening, overhearing, and over-emitting. For real-time data, a source node requests/forces a next-hop neighbor to wake up regardless to its normal schedule. The quick response from the intended target node almost eliminates the possibility of delay and over-emittance. Nodes also adopt their duty cycle according to the real-time request received from their neighbors.

Moreover, AREA-MAC is scalable and robust to topology changes. Unlike cluster-based approaches, where nodes only communicate via cluster heads, nodes using AREA-MAC communicate directly with peers.

2.2 Assumptions

We consider a grid-based WSN shown in Figure 2, consisting of several nodes and terminating at the sink node. This type of topology is useful in pre-deployment of many medical and surveillance related applications. All nodes except the sink node are normal nodes that sense, transmit, and receive. They have no aggregation or in-network capabilities. We assume that all nodes are fixed and know their locations regarding to some reference nodes. The selection and working of reference nodes is out of our scope. We assume that

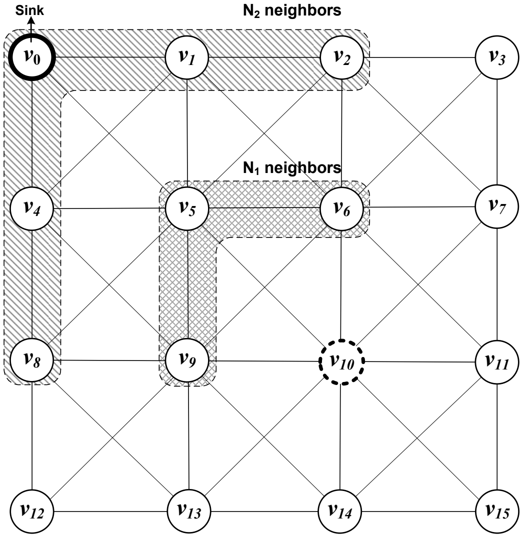

Figure 2: A portion of a grid-based WSN. All nodes are deployed in an ascending order with unique node IDs and know their locations (x,y). The deployment level decreases as the order or ID number increases.

the density of nodes is high enough, so that a node can directly communicate with multiple neighbors. All nodes carry unique node IDs and are deployed in an ascending order with a sink node having the highest deployment level. Normal nodes forward data only to up-level direction, i.e., towards the sink node. We also assume that all nodes except the sink node have limited and non-replicable energy resources.

We also define two terms called long-sleep and short-sleep. A node, after completing a communication with its neighbor, checks its data queue for any further data packet. If the queue is empty, it goes to a long-sleep mode and wakes up only after a complete slot duration. On the other hand, if the queue contains any data packet, the node remains in wake up mode and starts the process-cycle again (see Figure 1).

2.3 Network Model

A WSN is represented by an undirected graph $G(V, E)$, where $V = \{v_0, v_1, ..., v_{N-1}\}$ is the set of N sensor nodes' IDs and E is the set of edges connecting those nodes. Such graph can be described as a grid topology of $m \times n$ order with m rows and n columns. Nodes are placed at the location (x, y), where $1 \leq x \leq m$ and $1 \leq y \leq n$. Given the node ID, its location can be easily calculated and vice-versa [15]. The node v_0 represents the sink node, whereas nodes from v_1 to v_{N-1} represent normal sensor nodes. Table 1 shows all terms used for the network, energy, and delay models.

Table 1: Terms used for network, energy, and delay models

Symbol	Used for
N	Total number of sensor nodes
$N^1_{v_i}, N^2_{v_i}$	1-level and 2-level neighbors for v_i
T_χ, P_χ	Time and power in the respective state
$T_{a\chi}$	Average time in the respective state
T_{rxdata}, T_{txdata}	Time to RX/TX a data packet
$T_{rxhello}, T_{txhello}$	Time to RX/TX a hello packet
T_{rxpre}, T_{txpre}	Time to RX/TX a preamble
T_{byte}	Time to RX/TX one byte
T_{ack}	Time for an acknowledgement
T_{sw}	Time for radio switching
I_{wakeup}	Wake-up interval
I_{hello}	Hello interval

An up-level neighbor of a node v_i is called an 1-level neighbor for v_i, if its location parameters (x, y) satisfy one of these three conditions: (a) if its x value is equal to the x value of v_i, then its y value should be one less than y value of v_i, (b) if its y value is equal to the y value of v_i, then its x value should be one less than x value of v_i, or (c) both x, y values are one less than the x, y values of v_i. Similarly, an up-level neighbor of a node v_i is called a 2-level neighbor for v_i, if its location parameters (x, y) satisfy one of these three conditions: (a) its x value is two less than the x value of v_i, (b) its y value is two less than the y value of v_i, or (c) both x, y values are two less than the x, y values of v_i. For example, in Figure 2, the highlighted $N^1_{v_{10}} = \{v_5, v_6, v_9\}$ and $N^2_{v_{10}} = \{v_0, v_1, v_2, v_4, v_8\}$ are N_1 and N_2 neighbors for N_{10}. Each node $v_i \in V$ has a limited transmission range, but due to higher density, it can easily communicate with all of its $N^1_{v_i}$ and $N^2_{v_i}$ neighbors.

2.4 Energy Model

We divide the system time T in small discrete time intervals, $t_0, t_1, ..., t_n$. For simplicity, all time intervals are normalized to one time unit. The total energy consumption of a node v_i per unit of time, E_{v_i}, is given by its energy consumption in LPL, carrier sense, environment sense, reception, transmission, and sleep states respectively and shown in (1). Equation (2) shows the power consumption and the time spent by a node in the respective state for the respective time interval.

$$E_{v_i} = E_{lpl} + E_{carrier} + E_{sense} + E_{rx} + E_{tx} + E_{sleep} \tag{1}$$

$$\begin{aligned} &= P_{lpl}T_{lpl} + P_{carrier}T_{carrier} + P_{sense}T_{sense} \\ &\quad + P_{rx}T_{rx} + P_{tx}T_{tx} + P_{sleep}T_{sleep} \end{aligned} \tag{2}$$

A node performs LPL at every wake-up interval and senses the carrier before sending a preamble. It also senses the environment to measure physical values such as temperature, humidity, air velocity, or light.

$$T_{lpl} = \frac{T_{alpl}}{I_{wakeup}} \tag{3}$$

$$T_{carrier} = T_{acarrier}R_{data} \tag{4}$$

$$T_{sense} = T_{asense}R_{sense} \tag{5}$$

Here R_{data} is the rate at which a node sends and receives data packets and R_{sense} is the rate at which it senses the environment. The transmission time of a node is the sum of time required to send data packets, preambles, hello packets, and acknowledgement. At every hello interval, a node sends a hello packet. Whenever, it has data to send, it sends a preamble and immediately changes its radio to listen mode in order to receive the acknowledgement. This process is repeated until it receives an acknowledgement from the target node. Q is the number of attempts a node sends a preamble and changes its radio to receive an acknowledgement and P is the maximum number of allowed attempts, such that $0 \leq Q \leq P$.

$$T_{tx} = T_{txdata} + Q\left(T_{txpre} + T_{sw}\right) + T_{ack} + T_{txhello} \quad (6)$$

$$T_{txdata} = L_{data} R_{data} T_{byte} \quad (7)$$

$$T_{txhello} = \left(L_{hello}/I_{hello}\right) T_{byte} \quad (8)$$

Here L_{data} and L_{hello} are the total length of data and hello packet in bytes respectively. A node may receive multiple preambles during a time period. But when it becomes a target node for an specific preamble, it immediately sends an acknowledgement to the sender and receives its packet.

$$T_{rx} = T_{rxdata} + \sum T_{rxpre} + 2\, T_{sw} + T_{ack} + T_{rxhello} \quad (9)$$

$$T_{rxdata} = L_{data} R_{data} T_{byte} \quad (10)$$

$$T_{rxhello} = \left(L_{hello}/I_{hello}\right) T_{byte} \quad (11)$$

A node is supposed to be in sleep mode, if it is not doing anything else.

$$T_{sleep} = 1 - \left(T_{lpl} + T_{carrier} + T_{sense} + T_{rx} + T_{tx}\right) \quad (12)$$

2.5 Delay Model

The duty-cycle results in higher latency for WSNs. In AREA-MAC, all nodes are fixed and know their up-level neighbors. Therefore, in case of a real-time data, a node requests/forces an up-level neighbor to wake up regardless of its normal schedule and to perform data processing. For such a real-time data, a node wakes up aperiodically. We calculate the wake-up interval for an aperiodic traffic on the basis of Poisson distribution and calculate the expected number of real-time events occuring in an interval. If the rate of occurrences within an interval is λ, then the probability that there are exactly k occurrences is given by (13). For every occurrence of k, a node wakes up and performs data processing.

$$f(k;\lambda) = \left(\lambda^k e^{-\lambda}\right)/k! \qquad k \geq 0 \quad (13)$$

The total delay required to transfer a packet from v_i to v_j, i.e., D_{v_i,v_j} can be divided into three steps; delay at the source node v_i, delay at all intermediate nodes v_f, and delay at the destination node v_j, each denoted by D_{v_i}, D_f, and D_{v_j} respectively. If the set F contains all forwarding nodes such that $v_f \in F \subset V$, the processing delay at each node is T_{pro}, and the queuing delay is T_{que} then:

$$D_{v_i} = T_{lpl} + T_{sense} + T_{carrier} + T_{tx} + T_{sw} + T_{pro} \quad (14)$$

$$D_f = \sum_{v_f \in F} T_{lpl} + T_{carrier} + T_{rx} + T_{tx} + T_{sw} + T_{pro} + T_{que} \quad (15)$$

$$D_{v_j} = T_{lpl} + T_{carrier} + T_{rx} + T_{sw} + T_{pro} + T_{que} \quad (16)$$

$$D_{v_i,v_j} = D_{v_i} + D_f + D_{v_j} \quad (17)$$

3 CROSS-LAYERING WITH AREA-MAC

Though AREA-MAC is mainly associated with the MAC layer, it exploits simple routing at the network layer and interacts with application as well as PHY layers. Hence, the cross-layering with AREA-MAC is achieved between MAC, application, network, and PHY layers.

3.1 MAC and Application Layer

All nodes except the sink node generate data packets at the application layer and send them to lower layers. In this study, we use two type of traffic generating scenarios, simple and burst. In the simple traffic scenario, a node generates a random data packet at the rate of 30 seconds with the deviation of 30 seconds. Whereas, in the burst traffic scenario, a node generates double amount of data packets with the same rate and deviation. It means, for a simulation run of 1000 seconds, nodes generate on average 30 to 35 packets for the simple traffic scenario and 60 to 70 data packets for the burst traffic scenario. We examine average end-to-end delay, energy consumption, and packet delivery ratio of both traffic scenarios.

3.2 MAC and Network Layer

We consider three different type of simple yet effective routing schemes namely N_0, N_1, and N_2 routing [14]. In N_0 routing, nodes simply broadcast a packet and any neighbor node can receive and process the packet irrespective of its location. In N_1 routing, nodes send packets only to the up-level N_1 neighbors. Whereas, in N_2 routing, nodes send packets only to the up-level N_2 neighbors. We evaluate average end-to-end delay, energy consumption, and packet delivery ratio of all three routings for both simple and burst traffic generating scenarios.

3.3 MAC and PHY Layer

In order to save more energy, sensor nodes remain in sleep mode for most of the time. They wake up only at their wake-up time. This duty cycling makes the PHY layer very important for evaluating any MAC protocol for WSNs. Therefore, an ideal MAC protocol has always a strong coupling with the PHY layer. AREA-MAC periodically interacts with the PHY layer in order to distinguish between different radio modes and to control them efficiently. Moreover, to make things closer to the reality, suitable radio, channel, fading and noise models are used.

Table 2: Simulation configuration at different layers

Layer	Parameter	Value
General	Simulation area	800 × 800
	No. of nodes	16
	Topology	Grid
	Bit rate	19200 bps
	Simulation runs	25
	Simulation time/run	1000s
App layer	Packet payload	128 bits
	Data generating interval	30s
	Data generating deviation	30s
	Burst traffic rate	2
Network layer	Routing	N_0, N_1, N_2
MAC layer	Packet header length	10 bits
	Queue size	100
	Slot duration	5s
PHY layer	Transmission power	0.05mW
	Carrier frequency	868MHz
	Propagation model	Nakagami

4 PERFORMANCE EVALUATION

AREA-MAC is implemented on the basis of an OMNeT++ based simulation framework given in [14]. The simulation configuration used for the framework is given in Table 2. All results are drawn by using the box-whisker graphs, where rectangular boxes show the confidence interval of 95% and the line within boxes shows the median. The upper and lower whisker bars show the maximum and minimum values respectively. We discuss delay, energy consumption, and packet reception rate first for the simple traffic generating scenario and then for the burst traffic scenario.

4.1 Simple Traffic Scenario

4.1.1 Average end-to-end delay

The average end-to-end delay is the average of times per simulation run needed for data packets to arrive at the sink node. Figure 3 shows the average end-to-end delay for all three type of routings. This figure suggests that the N_1 and N_2 routing significantly decrease the delay at the sink node compared to the N_0 routing. With N_1 and N_2 routing, nodes only send/forward packets in the direction of the sink node, i.e., towards the up-level neighbors. This decreases the average end-to-end delay by almost half for the N_1 routing as compared to the N_0 routing. Moreover, the N_2 routing further improves this delay where nodes send data directly to their 2-level neighbors.

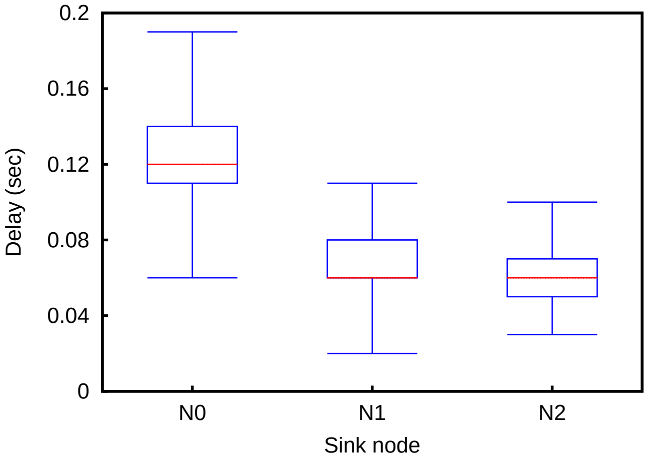

Figure 3: Delay at the sink node

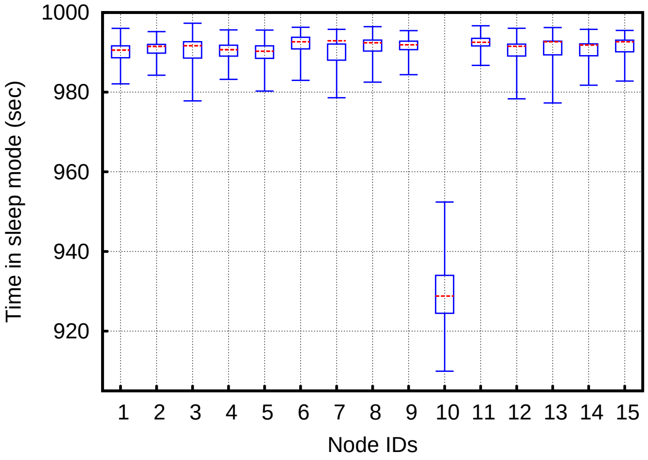

Figure 4: Time spent in sleep mode

4.1.2 Energy consumption

We calculate energy consumption for a node by the time for which its radio remains in different modes. Figures 4, 5, and 6 show the time spent in sleep, receive, and transmit mode respectively for a node. The results achieved with our study show that for all three type of N_0, N_1, and N_2 routings, times spent in sleep, receive, and transmit modes for a node are almost same. Therefore, we show here sleep, receive, and transmit time for nodes for only N_0 routing. The N_1 and N_2 routing improves the overall energy consumption of a network by

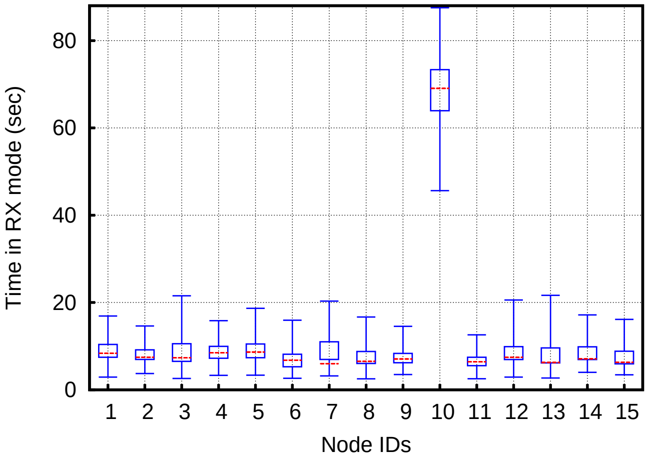

Figure 5: Time spent in receive mode

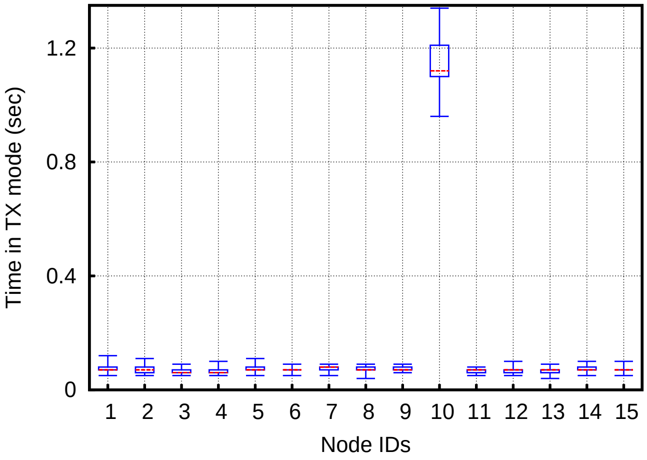

Figure 6: Time spent in transmit mode

only a small factor. For an experiment run of 1000 seconds, average sleep times for a node for N_0, N_1, and N_2 routing are 986, 987, and 988 seconds respectively. However, for a WSN, which is usually deployed for a longer period of time, this marginal improvement further prolongs network lifetime. Thus, the duty cycle provided by AREA-MAC is more than 98%.

4.1.3 Packet delivery ratio

In AREA-MAC, all nodes except the sink node generate, receive, and forward data packets to the sink node. Hence, packets sent by a node include both the packets it generates and receives from its neighbors. Figure 7 shows the packet delivery ratio of the packets received at the sink node to the packets generated by other nodes. This ratio is higher for the N_0 routing as compared to the N_1 and N_2 routings. This is because of the fact that nodes using N_0 routing can receive packets from any of their neighbors irrespective to their level and location. Whereas, in N_1 and N_2 routing nodes only receive packets from their low-level neighbors and forward them to their up-level neighbors.

4.2 Burst Traffic Scenario

4.2.1 End-to-end delay

Figure 8 affirms that the end-to-end delay for data packets decreases with the N_1 and N_2 routing. The N_1 routing roughly improves delay by half of the N_0 routing, which is further improved by the N_2 routing. This is because of the same fact that with N_1 and N_2 routing, nodes only send/forward packets towards up-level neighbors, whereas with N_0 routing nodes simply broadcast packets and hence packets may be received by low-level neighbors, i.e., in the opposite direction of the sink node.

4.2.2 Energy consumption

Similar to the simple traffic scenario, times spent in sleep, receive, and transmit mode for a node in each of the routing category are almost same for the burst traffic scenario. Therefore, we show here times spent in sleep, receive, and transmit modes by each node for only N_0 routing in Figures 9, 10, and 11 respectively. Results prove that the burst traffic generation does not really effect the energy consumption and end-to-end delay for a network using AREA-MAC.

4.2.3 Packet delivery ratio

For the burst traffic scenario, nodes generate on average 60 to 70 data packets. As learned from the simple traffic scenario, packets delivery ratio at the sink node is higher with N_0 routing than N_1 and N_2 routing. Figure 12 shows that still more than 85% of packets are successfully received by the sink node.

5 RELATED WORK

MAC protocols for WSNs can be classified into the two broad categories of *contention-based* and *schedule-based* protocols. In contention-based MAC protocols nodes compete to acquire the channel. Whereas, in scheduling-based protocols, a schedule assigns a time slot and resources to a node. We briefly discuss here some of the well-known MAC protocols from both categories and then present state of the art related to the cross-layering for WSNs. The detailed discussion of different MAC protocols for WSNs is given in [15].

S-MAC [1] circumvents idle listening, collisions, and overhearing by using periodic and fixed-length wake-up and sleep periods, but it is rigid and optimized for a predefined set of workloads. Synchronization and longer sleep periods result in higher latency. Time-out MAC (T-MAC) [7] protocol improves S-MAC by adaptively shortening the listen period by monitoring for a threshold period. T-MAC suffers from early sleeping, reduced throughput, additional latency, complexity, and scaling problems. The CSMA based B-MAC [9] protocol

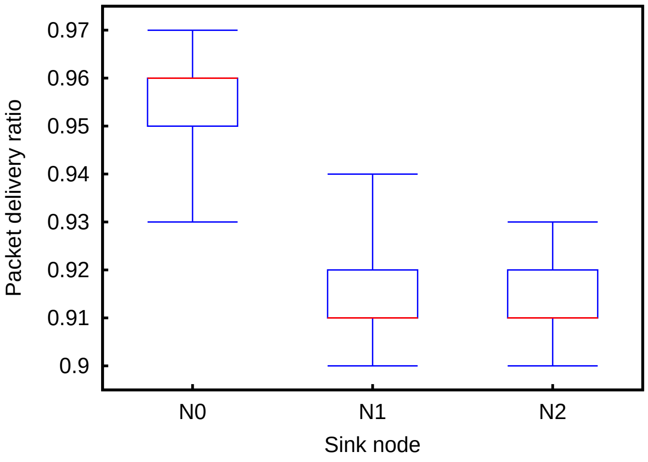

Figure 7: Packet delivery ratio

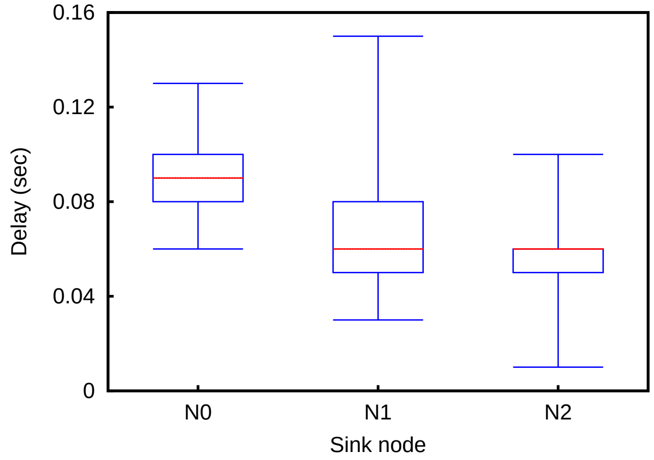

Figure 8: Burst traffic - delay at the sink node

uses low power listening (LPL) with an extended preamble to reduce duty cycle and idle listening. It has an overhearing issue and the long preamble dominates the energy usage. The TDMA based LEACH [2] divides a WSN into clusters, each supervised by the cluster-head. Always-on cluster-heads and scalability are major problems with LEACH. The IEEE 802.15.4 standard [16] provides several features for WSNs. But it also carries limitations, especially for real-time, energy-efficient, and bandwidth critical WSN applications, identified in [17].

The authors in [10] explains the importance of cross-layer design for WSN, especially for their stringent energy, storage, and processing capabilities. Radio, wireless channel, and application-aware protocols definitely improve overall WSN performance. However, they also discuss concerns and precautionary considerations regarding cross-layer design architecture for WSN. An analytical energy survey for three different MAC protocols for physical, data link, and network layer is proposed in [12]. They argue that single-hop communication has up to 40% lower energy consumption than multi-hop forwarding within the feasible transmission distance. In [18], a cross-layer optimization of network throughput for multi-hop wireless networks is spilted into two sub-problems, one for multi-hop flow routing at the network layer and the other for power allocation at the physical layer. The authors claim that their framework may handle the throughput optimization problem in an efficient and distributed fashion for a broad range of wireless network scenarios. A MAC framework is combined with routing in [13] in order to achieve higher energy-efficiency in WSNs.

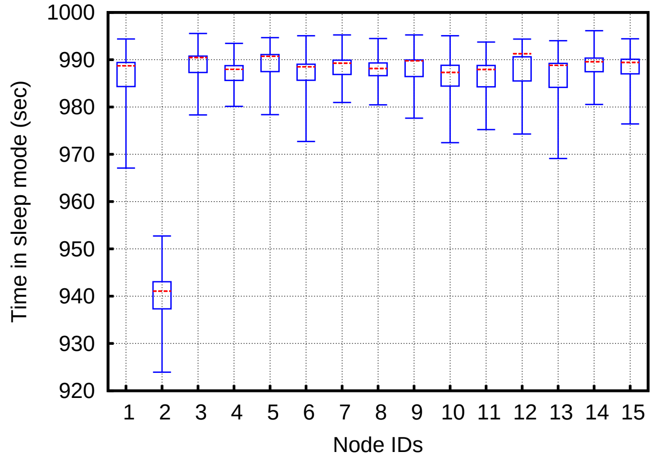

Figure 9: Burst traffic - time spent in sleep mode

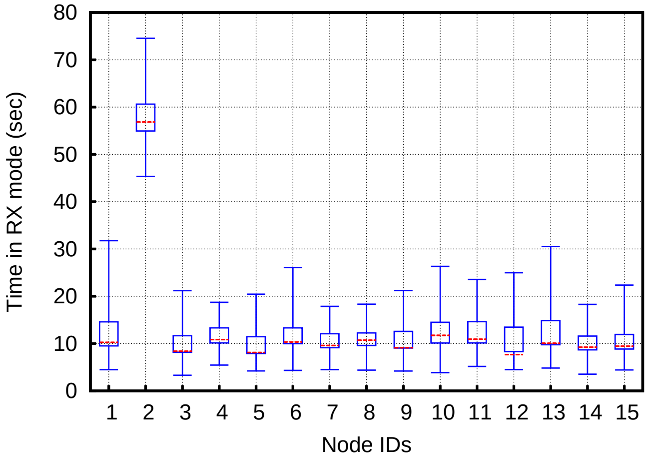

Figure 10: Burst traffic - time spent in receive mode

Accordingly, a sufficient amount of research is available which focuses on the cross-layer designing for WSN. However, a negligible amount of work is archived for delay-bound WSN applications. This motivates us to propose, design, and evaluate a new MAC protocol called AREA-MAC.

6 CONCLUSION

This paper evaluates AREA-MAC with the perspective of cross-layer design. AREA-MAC is an asynchronous and adaptive MAC protocol which deals with time and energy critical WSN applications. It also interacts with network, applica-

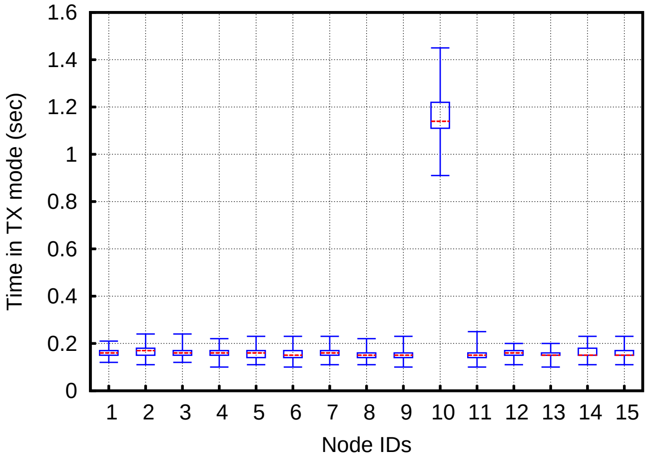

Figure 11: Burst traffic - time spent in transmit mode

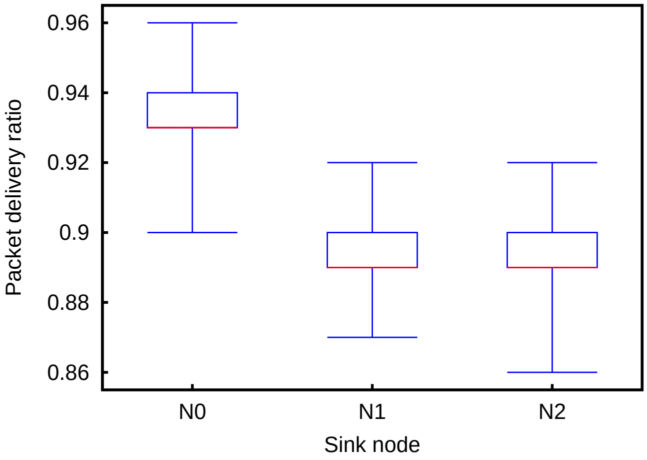

Figure 12: Burst traffic - packet delivery ratio

tion and PHY layers to enhance overall network performance. This simulation study evaluates AREA-MAC for two different type of traffic generating scenarios, i.e., simple and burst. We also use three different types of routing schemes namely N_0, N_1, and N_2. Moreover, AREA-MAC has a strong coupling with the PHY layer in order to control the switching of radio from one mode to other. Results show that the cross-layering in AREA-MAC decrease end-to-end delay and energy consumption of the network, but it slightly decreases the packet reception ratio at the sink node. Our next task is to convert these results into some general mathematical formulas and then optimize end-to-end delay and energy consumption by varying the duty cycle with the help of linear programming.

REFERENCES

[1] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Net.*, 12(3):493–506, June 2004.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), October 2002.

[3] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *ACM SenSys, Los Angeles, CA*, November 2003.

[4] A. El-Hoiydi and J. Decotignie. WiseMAC: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. *Ninth IEEE Symposium on Computers and Communication, ISCC04*, pages 244–251, June 2004.

[5] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE transactions on Mobile Computing*, 1(1):70–80, 2002.

[6] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. *IEEE WCNC*, 3:1534–1539, March 2004.

[7] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 171–180, November 2003.

[8] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A short preamble mac protocol for duty-cycled wireless networks. *4th ACM Conf. on Embedded Networked Sensor Systems (SenSys), Boulder, CO*, pages 307–320, November 2006.

[9] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. *2nd ACM Conf. on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD*, pages 95–107, November 2004.

[10] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili. The State of the Art in Cross-layer Design for Wireless Sensor Networks. In *Proceedings of EuroNGI Workshops on Wireless and Mobility. Springer Lecture Notes in Computer Science 3883*, Como, Italy, July 2005.

[11] Vikas Kawadia and P. R. Kumar. A Cautinary Perspective On Cross-Layer Design . In *IEEE Wireless Communication Magazine*, February 2005.

[12] Jussi Haapola, Zach Shelby, Carlos Pomalaza-Raez, and Petri Mahonen. Cross-Layer Energy Analysis of Multi-hop Wireless Sensor Networks. In *EWSN'05*, pages 33–44, 2005.

[13] Lodewijk Van Hoesel, Tim Nieberg, Jian Wu, and Paul J. M. Havinga. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. In *IEEE wireless communication*, volume 11, pages 78–86, December 2004.

[14] Pardeep Kumar, Mesut Güneş, Qasim Mushtaq, and Bastian Blywis. A real-time and energy-efficient MAC protocol for wireless sensor networks. *International Journal of Ultra Wideband Communications and Systems (IJUWBCS)*. to be published, `http://sites.google.com/site/ijuwbcs/accepted`.

[15] Pardeep Kumar and Mesut Güneş. MAC protocols for low-latency and energy-efficient WSN applications. Technical Report TR-B-08-18, FU Berlin, Germany, December 2008.

[16] IEEE 802.15.4. MAC and PHY specifications for LR-WPANs. 2006. `http://ieee802.org/15/pub/TG4.html`.

[17] Pardeep Kumar, Mesut Güneş, Abd Albasset Almamou, and Jochen Schiller. Real-time, bandwidth, and energy efficient IEEE 802.15.4 for medical applications. *7. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", FU Berlin, Germany*, September 2008.

[18] Jun Yuan, Zongpeng Li, Wei Yu, and Baochun Li. A cross-layer optimization framework for multicast in multi-hop wireless networks. In *WICON'05*, pages 47–54, July 2005.